教育資料

パスカル・ファイル・システム入門 (1)

藤田 芳夫*

コンピュータの学習で必要欠くべからざるものにファイルの操作がある。どんな分野を専攻するにせよ、ファイルの処理、操作の方法を十分に理解することは必要であるが、コンピュータ会計、管理会計の分野を目指す学生にとってファイルの処理法、操作の原理を理解することは絶対に欠くことの出来ない必要条件である。

この意味で本稿はターボ・パスカルによるランダム・アクセス・ファイルの操作法の初歩を以下の項目に分けて丁寧に解説した。

＊東京情報大学教授

1992年2月18日受理

第1章　整数ファイル …最も単純なランダム・アクセス・ファイル…

1-1. 整数ファイル作成プログラム intfile. pas

テキスト・ファイルは順アクセス・ファイルであるのに対し、ファイルを構成する任意のレコードに直接アクセス出来るランダム・アクセス・ファイルの最も単純な例として、ここでは各レコードが整数（インデジャー）1個だけからなるファイル fx : file of integer を考えてみることにする。

いま、ファイルの先頭レコードは $(-30)^{\wedge}3 = -27000$、次は $(-29)^{\wedge}3 = -24389$、…、ファイルの末尾は $(+30)^{\wedge}3 = 27000$からなる61個の整数を書き込んだファイル “m : intfilc. dat” を作ってみよう。

このプログラムは次のようにすればよい。

```
program intfile;
  var i, n : integer;
        fx : file of integer;
  begin
  assign(fx,'m : intfile. dat');
  rewrite(fx);
  for i : = −30 to 30 do
   begin
    n : = i * i * i;
    write(fx, n);
   end;
   close(fx);
  end.
```

このプログラムで大事な点は

1) 変数宣言部でファイル変数 fx を file of integer として宣言していること。

2) プログラム実行部の先頭で、ファイル実体名とドライブ名をファイル変数 fx に割り当てていること。

```
assign(fx, 'm : intfile. dat');
```

3) 61個の変数値を持つファイル 'm : intfile. dat' を始めて作るのであるから、

rewrite(fx) としている点。

4) 61個の整数データをファイルに書き込むため、for ループのなかで write(fx, n)としている点。

5) そして最後に close(fx) としている点である。

この最後のクローズ文は作成したデータ・ファイルの情報がフロッピーのディレクトリー関係部分に記入されると言う点で重要な操作であるから、これを忘れてはならない。

(問) 上掲プログラムを実行しなさい。

1-2. 整数ファイル読みだしプログラム rdifil1. pas

次に、前節で作成した整数ファイル 'm : intfile. dat' の内容を書き出すプログラム rdifl1. pas(read intfile. dat) を考えてみよう。

ここでは、フロッピーに作成されている 'intfile. dat' を読み出すのであるから、

1) 変数宣言部でファイル変数 fx が整数を記録単位とするファイルであるという宣言をする。これが fx : file of integer である。

2) 次に実行部分の先頭でファイル実体名 'm : intfile. dat' を fx に割り当てる。

```
assign(fx,' m : intfile. dat')
```

3) そして既存のファイルを読み出すのであるから、reset 文でファイルをオープンする。

```
reset(fx)
```

4) オープンしたデータ・ファイルの末尾に至るまで、整数データ一個をファイルから読みだし、CRT 画面に書き出す作業を繰り返す。これが下に示す while ループである。

```
while not eof(fx) do
  begin
    ファイルから整数一個を読み出して書く
  end;
```

ここでファイルの末尾に到達したか否かを eof(fx) 関数で判定しているのである。

5) 'm : intfile. dat' に含まれている61個の整数データを見やすくするため、一行に八個書いた

ら改行するため、
if(j mod 8)=0 THEN writeln ; を使用している。
6) 作業が終了すると、クローズ文でファイルをクローズしている点に注意しなければならない。

```
program rdifl1 ;
 var fx : file of integer ;
     i, j : integer ;
 begin
  assign(fx, 'm : intfile. dat') ;
  reset(fx) ;
  j : =1 ;
 while not eof(fx) do
  begin
   read(fx, i) ;
   write(i : 7) ;
   if (j mod 8)=0 then writeln ;
   j : =j+1 ;
 end ;
close(fx) ;
end.
```

(問) 'intfile. dat' の全データを書き出すためには、while not eof(fx) do と言う while ループを使うほか、filesize(fx) 関数を使って、次のようにしてもよい。

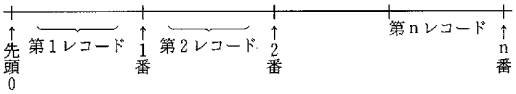
```
for i : =1 to filesize(fx) do
  begin
    ｛
  end ;
```

この方法を使って 'intfile. dat' の全データを書き出すプログラムを作成し、実行しなさい。

1-3. ランダム・アクセス・プログラム raccessl. pas

上述したように、ランダム・アクセス・ファイルは特定のレコードに直接アクセスする事が出来る。このランダム・アクセスはレコード・ナンバー (n) を与えて seek(fx, n-1) 手続きで行う。

形つきファイル（すなわちランダム・アクセス・ファイル）のレコード・ナンバーはファイルの先頭位置がゼロ番で、以下レコードの順序数はそのレコードの末尾を指すことになる。



したがって、第n番レコードを読み出そうとすれば、seek(fx, n-1) としなければならない。

以上の点を考慮してプログラミングすれば、下のようになる。

```
program raccessl ;
 var fx : file of integer :
  rn,          {レコード・ナンバー          }
 rec : integer ;{intfile. dat 上の整数レコード }
 begin
  assign(fx, 'm : intfile. dat')
  reset(fx) ;
  write ('レコード・ナンバー=>') ;
  readln(rn) ;
  while rn in  [1..filesize(fx)]  do
   begin
    seek(fx, rn-1) ;
    read(fx, rec) ;
    writeln('              =', rec)
    write(' レコード・ナンバー=>') ;
    readln(rn) ;
   end ;
  close(fx) ;
  end.
```

なお、このプログラムでは1番から61番以外のレコード・ナンバーを入力すれば、プログラムは停止するように作成してある。この1番から61番までのレコード・ナンバーかどうかを判定するのが

```
while rn in  [1..filesize(fx)]  do
begin
  ｛
end ;
```

である。

関数 filesize(fx) はファイル変数 fx についてファイルに含まれているレコードの数を返すから、rn がファイルに含まれるレコード・ナンバーでありさけすれば、直ちにアクセスする事が出来るのである。
(問) 適当なレコード・ナンバーを入力して、プログラム raccessl. pas を実行しなさい。

1-4. データ更新プログラム rupdate. pas

通常、ファイルは作成後、構成レコードを修正または更新する必要が生まれる。たとえば、住所録では移転があると、その人の住所、そして電話番号も修正しなければならない。

ランダム・アクセス・ファイルではこの修正を容易に行うことが出来る。i 番レコードを修正(又は更新) するためには、

1. seek(fx, i-1) で i 番レコードの先頭に位置づけ、
2. read(fx,x) で i 番レコードの修正前の内容を x に読み込み、
   writeln(x) で i 番レコードを画面に表示し、
3. readln(x) で新レコードをキーボードから入力し、
4. seek(fx, i-1) で i 番レコードの末尾に移動しているファイル・ポインターを再び i 番レコードの先頭に位置づけ、
5. write(fx, x) で新しい i 番レコードを元の位置へ書き込んで更新する。

以上が、i 番レコードを更新する基本的考え方である。

この基本的な考え方を用いて整数ファイル 'm : intfile. dat' の任意の個数のレコード (すなわち整数) を変更するプログラムをつくると次のようになる。

```
program rupdate1;
  var fx : file of integer;
    rn, rec : integer;
    yes : char;
begin
  assign(fx, 'm : intfile. dat');
  reset(fx);
  write('record no=>'); readln(rn);
  while rn in  [1..filesize(fx)] do
    begin
      seek(fx, rn-1); ……
      read(fx, rec);                    } 旧データの読
      writeln('        =', rec);        } 出しと表示
      write(' 新データ入力=>'); readln(rec);
            ……新データ入力
      write('Y/N          =>'); readln(yes);
            ……新データのチェック
      if yes in  ['Y', 'y']  then…      }
        begin                           } 新データ
          seek(fx, rn-1);               } による
          write(fx, rec);               } 更新
        end;          ……               }
      write('record no.=>'); readln(rn);
    end;
  close(fx);
end.
```

(問) 第 1 レコードを32768、第10レコードを12345、第55レコードを11111に変更しなさい。
(問) 上の問題で修正された整数ファイルの全内容を rdifl1. pas で表示し、予想通り修正されていることを確認しなさい。

1-5. データ追加プログラム apdifl1. pas

前節までで整数ファイルの作成、表示、ランダム・アクセス、更新等の方法について基本的な方法を考察した。本節ではデータの追加についてプログラムを考えてみよう。

ファイルの末尾に新しいデータを追加するためには、
seek(fx, filesize(fx)) でファイル末尾を出し、
write(fx, rec)　　で書き込めばよい。複

数データがあるときは、このwrite文を繰り返すのである。最後にclose文で締めくくる事は言うまでもない。

そこでプログラムは次のようになる。

```
program apdifl1;
 var fx: file of integer;
       i: integer;
   yes: char;
  begin
   assign(fx,' m: intfile. dat');
   reset(fx);
   seek(fx, filesize(fx));

   repeat
    write('ツイカデータニュウリョク->');
    readln(i);
    write(fx, i);
    write('ツイカケイゾク Y/N->');
    readln(yes);
   until upcase(yes)<>'Y';
   close(fx);
  end.
```

このプログラムの要点は、seek(fx, filesize(fx)) で末尾を一回だけ出し、追加データはrepeat……until文で、追加を継続するか否かを示す変数yesがノーとなるまでデータの追加を繰り返す点である。

(問) 新データ2345と22222を追加し、その結果をrdifl1. pasで確認しなさい。

以上、整数データ・ファイルと言う極端に単純なレコードを用いてランダム・アクセス・ファイルの特徴を説明した。このほかにも、レコードの削除やレコード番号によるレコードの探索ではなく、レコードの値（例えば、人名）によってファイルから特定のレコードを探す方法を明らかにしなければならないが、これらの点は、より標準的なレコード（たとえば住所録とか図書カード・ファイル）を用いて説明するほうがよいので、次章以下にゆずることにする。

第二章 住所録ファイル

前章で説明したランダム・アクセス・ファイルとしての整数ファイルはレコードの構造が単純であったので、本章ではレコードとしてより標準的な住所録をとりあげ、ランダム・アクセス・ファイルの説明をすることにする。

2-1. 住所録作成プログラム stdfilel. pas

住所録の単純な形は氏名、住所、電話番号からなっており、各フィールドは

氏名……20バイト
住所……40バイト
電話番号……14バイト

であるとする。

10人分の住所録を作成するためには、まず第一に、タイプ宣言でファイルを構成するレコードをstdrec(standard record)として宣言する。

```
type stdrec=record
              n: string[20];……氏名
              a: string[40];……住所
            tel: string[14];……電話番号
    end;
```

ファイル変数fxはこのstdrecを構成要素とするファイルであるから、

```
var fx: file of stdrec;
```

と宣言し、プログラムのなかで変数として使用するレコードは

```
rec: stdrec;
```

として宣言するのである。

以上の準備の後、次のassign文とrewrite文でファイルを開くのである。

```
assign(fx,'m: stdfilel. dat');
rewrite(fx);
```

10人の氏名、住所、電話番号は下のfor文で記録する。

```
for i:=1 to 10 do
  begin
  氏名のプロンプト；氏名の入力；
```

```
        住所のプロンプト；住所の入力；
        電話番号のプロンプト；電話番号の入
                                力；
    end;
```

プログラム全体をまとめて示すと、次の通りである。

```
program stdfilel;
 type stdrec=record
                n:string[20];
                a:string[40];
              tel:string[14];
             end;
var fx:file of stdrec;
    i:integer;
  rec:stdrec;
 begin
  assign(fx,'m:stdfile.dat');
  rewrite(fx);
  for i:=1 to 10 do
   begin
    write('rec. n=');readln(rec.n);
    write('rec. a=');readln(rec.a);
    write('rec. tel=');readln(rec.tel);
    write(fx, rec);
   end;
 close(fx);
end.
```

住所録を始めて作成するとき、データが十人分に固定するのを避けるためには、次のプログラム stdfile2. pas のようにすればよい。

```
program stdfile2;
 type stdrec=record
          n:string[20];
          a:string[40];
        tel:string[14];
          end;
 var
  fx:file of stdrec;
 rec:stdrec;
begin
  assign(fx,'m:stdfilel.dat');
  rewrite(fx);
  repeat
   write('氏名->');readln(rec.n);
   write('住所=>');readln(rec.a);
   write('電話=>');readln(rec.tel);
   write(fx, rec);
  until rec.n[0]=chr($0);
  close(fx);
 end.
```

ここでは、入力データの打ち切りは、氏名を入力するさい、キャリッジ・リターン・キーだけをキー・インすれば rec. n[0] に16進のゼロが入るので

until rec. n[0]=chr($0) で、氏名が空文字列であることが判定され、ファイルの作成を中止するのである。

(問) stdfilel. pas 又は stdfile2. pas を使って住所録を作成しなさい。

2-2. 住所録ファイルの書き出しプログラム rdstdfl. pas

作成した住所録の全記録をプリンターに出力するプログラムは次のようになる。

```
program rdstdfl;
 uses printer;
 type stdrec=record
               n:string[20];
               a:string[40];
             tel:string[14];
           end;
 var fx:file of stdrec;
  rec:stdrec;
    i:integer;
begin
 assign(fx,'m:stdfile.dat');
 reset(fx);
 i:=1;
```

```
while not eof(fx) do
 begin
  read(fx, rec);
  writeln(lst,' レコード・ナンバー=', i);
  writeln(lst,'      氏名=', rec. n);
  writeln(lst,'      住所=', rec. a);
  writeln(lst,' 電話番号=', rec. tel);
  writeln(lst);
  i:=i+1;
 end;
close(fx);
end.
```

ここでは、既存の住所録を読むだけであるから、reset(fx) を用いる事。またファイル全体を書き出すのであるから、

```
while not eof(fx) do
begin
 1レコードを読みだし
 そのレコードを書き出す
end;
```

と言う構造にしたのである。

変数 i はレコードの読みだし順に対応して、レコード・ナンバーをつけるためである。

なお、1レコードを読み出すには、

```
read(fx, rec)
```

でよく、read(fx, rec. n, rec. a, rec. tel);
とする必要はない。

読みだしたレコードの氏名 (n)、住所 (a)、電話番号 (tel) をプリンターに書き出すためには、各フィールド毎に説明をつけ、また rec. n, rec. a, rec. tel のように、するのである。

(問) プログラム rdstdfl. pas を実行しなさい。

2-3. レコード・ナンバーで直接アクセスするプログラム rastdfl1. pas

前節で展開したプログラムは住所録全体をプリント・アウトしたが、ファイルが大きくなると、指定したレコードだけを読み出すことが出来なければ困る。この点を解決するのが本節のプログラム rastdfl1. pas であり、整数ファイルのランダム・アクセスを行ったプログラム raccessl. pas と基本的に同じ構造をしている。

```
program rastdfl1;
 type stdrec=record
              n:string[20];
              a:string[40];
            tel:string[14];
           end;
var rn:integer;
    fx:file of stdrec;
   rec:stdrec;
begin
 assign(fx,'m:stdfile. dat');
 rese(fx);

 write('レコード・ナンバー>');
 readln(rn);
 while rn in [1..filesize(fx)] do
  begin
   seek(fx, rn-1);
   read(fx, rec);
   writeln('氏名=', rec. n);
   writeln('住所=', rec. a);
   writeln('電話=', rec. tel);
   writeln;
   write('レコード・ナンバー>');
   readln(rn);
  end;
 close(fx);
end.
```

このプログラムの特徴は seek(fx, rn-1) にあることは整数ファイルのランダム・アクセスの場合と同じである。またレコードの読みだしと、レコード各部の書き出しについては、前節のファイル読みだしプログラム rastdfl. pas の場合と同じである。

(問) 任意のレコード・ナンバーを入力して特定の人物の住所を出力しなさい。

2-4. 住所録更新プログラム updstdfl. pas

ファイルは生き物であって、いちど作成したファイルは、ほとんどの場合、更新する必要が生まれる。

ある人の住所や電話番号が変わったり、住所録から削除する必要が起きたり、新しい人名を追加する必要が生まれる。

特定レコードの内容を変更する場合、まずそのレコードにアクセスしなければならないが、その際、レコード・ナンバーでアクセスする方法と、特定フィールドの内容、例えば氏名でアクセスする方法がある。氏名でアクセスする方法については後章に譲り、ここではレコード・ナンバーでアクセスする単純な方法を考えることにする。

このレコード・ナンバーでアクセスして更新するプログラムの内容は整数ファイルの更新プログラム rupdate. pas と基本的に同じで、レコード番号 rn のレコードを

seek(fx, rn-1);……で探し、そのレコードの内容を

read(fx, rec); ……で読み

```
writeln('氏名=', rec. n);
writeln('住所=', rec. a);
writeln('電話=', rec. tel);
writeln;
```

で氏名、住所、電話番号を書き出し

次に、更新すべき内容を入力し、その正否を確認し、更新してよいと確認したものだけを、

```
seek(fx, rn-1);
write(fx, rec);
```

でそのレコードの位置へ書き込むのである。

```
program updstdfl;
 type stdrec=record
                n: string[20];
                a: string[40];
              tel: string[14];
             end;
  var rn: integer;
      fx: file of stdrec;
     yes: char;

begin
 assign(fx,' m: stdfile. dat');
 reset(fx);

 write(レコード・ナンバー=>);
 readln(rn);
 while rn in [1..filesize(fx)] do
  begin
   seek(fx, rn-1);
   read(fx, rec);
   writeln('Rec. No.=', rn);
    writeln('氏名=', rec. n);……
    writeln('住所=', rec. a);……   旧データの書き出し
    writeln('電話=', rec. tel);……
        writeln;
   writeln(' 新データ入力->');
   writeln('氏名='); readln(rec. n);
   writeln('住所='); readln(rec. a);   新データの入力
   writeln('電話='); readln(rec. tel);
    writeln('確認 Y/N->'); readln(yes);
                                  新データの確認
    if upcase(yes)='Y' then
     begin                        確認データの書込
     seek(fx, rn-1);
     write(fx, rec);
     end;
  write(' レコード・ナンバー=>');
  readln(rn)
  end;
  close(fx);
 end.
```

このプログラムでレコードを削除するときは、新データの入力要求に対して、氏名、住所、電話番号の全てに対して空文字列を入力することにする。なお、削除については後で改良する。

レコードの更新について、このプログラムでは、例えば電話番号だけ修正する場合でも、氏名、住所について同じ氏名、住所を入力しなけ

ればならない、と言う不合理さを持っている。この点も、後に改良することにする。

(問) 住所録レコードの全項目、および一部項目の修正を実行しなさい。

2-5. 住所録追加プログラム apdstdfl. pas

住所録に含まれているデータを更新するだけでなく、新しいデータを追加する必要も発生する。このデータ追加の方法も、整数ファイルの場合のデータ追加プログラムと殆ど同じで、

seek(fx, filesize(fx));

でファイルの末尾をだしておき、追加データの数だけ

write(fx, rec); を繰り返せばよい。

```
program apdstdfL;
 type stdrec=record
                n:string[20];
                a:string[40];
              tal:string[14];
             end;
  var
    fx: file of stdrec;
   rec: stdrec;
   yes: char;

  begin
   assign(fx,' m:stdfile. dat');
   reset(fx);
   seek(fx, filesize(fx));

  write(' 追加 Y/N ->'); readln(yes);
  while (upcase(yes)='Y') do
   begin
    write(' rec. n=       '); readln(rec. n);
    write(' rec. adress=' ); readln(rec. a);
    write(' rec. tel=     '); readln(rec. tel);
    write(fx, rec);
    write(' 追加 Y/N ->'); readln(yes);
   end;
  close(fx);
 end.
```

(問) 住所録に3人分のデータを追加しなさい。

第三章 住所録 管理 システム・プログラム

3-1. ファイル管理システム・プログラム stdfsys0. pas

前章で展開した五つのプログラムは基本的データ・ファイル 'stdfile. dat' を作成し、管理・維持するためのプログラムである。従って、これらのプログラムを一つにまとめておくと、使用するとき便利である。

そこで、住所録を最初に作成するときに使用したプログラム (stdfilel. pas又はstdfile2. pas) 以外、ファイルのメインテナンスのための四つのプログラムを一つのシステムにまとめる方法を考えてみる事にする。

その方法は、ファイルを構成する基本レコードのタイプ宣言と、メインテナンスの為、各プログラムで共通に使用する変数をくくりだし、各プログラムはそれらを外した部分を手続き (proceduer) として、メイン・プログラムから呼ぶことにするのである。

この基本的構造を示すと、次の stdfsys0. pas のようになる。このプログラム自体は、実行しても何もしないが、文法的なエラはなにもない。

このstdfsys0. pasの四つの手続きを一つづつ具体化して、正しく働くことを確認しつつ、システムを完成すればよいのである。

```
program stdfsys0;
 type stdrec=record
               n:string[20];
               a:string[40];
             tel:string[14];
            end;
 var rn: integer;
     fx: file of stdrec;
    rec: stdrec;
    yes: char;
```

```
      k : integer ;

procedure updstdfl ;
 begin
 end ;
procedure apstdfl ;
 begin
 end ;
procedure rastdfl ;
 begin
 end ;
procedure rdstdfl ;
 begin
 end ;

begin
 writeln(' 全 レコード ヲ Prn ニ出力…1') ;
 writeln(' Rn レコード ヲ Crt ニ表示…2') ;
 writeln('新 レコード ヲ ファイルニ追加…3') ;
 writeln(' レコード ノ 更新          …4') ;
 readln(k) ;
 case k of
  1 : rdstdfl ;
  2 : rastdfl ;
  3 : apstdfl ;
  4 : updstdfl ;
 end ;
end.
```

(問) プログラム stdfsys0. pas を実行しなさい。

3-2. 更新プログラムを手続き化した stdfsys1. pas

stdfsys0. pas を見れば明らかなように、最初に更新手続きみ置いたので、住所録ファイルを更新するプログラム updstdfl. pas を手続き化することから始めよう。

このため前章で展開した更新プログラム updstdfl. pas からレコード・タイプの宣言と変数宣言を取り除き、一行目の“program”を“procedure”に代え、最後の“end.”文のピリオドをセミコロンに代えるだけでよい。

なお、この作業を行うとき、プログラム updstdfl. pas を一文字づつキー・インする必要は無いのであって、次のような Turbo Pascal の便利な編集コマンドを使用するのである。

1．まず procedure　updstdfl と procedure apdstdfl の間にプログラム updstdfl. pas が入るだけの空白部分をあける。

2．次に、CTRL-K、CTRL-R とキー・インすると、デイスクから何を読み込むかを聞いてくるので、“m : updstdfL. pas”と答えればよいのである。

3．以上により、指定の場所に更新用プログラム“updstdfl. pas が読み込まれるので、不要な部分を削除し、必要な修正を加えればよいのである。

```
program stdfsysl ;
 type stdrec=record
                n : string[20];
                a : string[40];
              tel : string[14];
             end ;
 var rn : integer ;
     fx : file of stdrec ;
    rec : stdrec ;
    yes : char ;
      k : integer ;

procedure updstdfl ;
 begin
  assign(fx, 'm : stdfile. dat') ;
  reset(fx) ;

  write(レコード・ナンバー=>) ;
  readln(rn) ;
  while rn in [1..filesize(fx)] do
   bein
    seek(fx, rn-1) ;
    read(fx, rec) ;
   writeln('Rec. No. =', rn) ;
```

```
    writeln('名前    =', rec. n);
    writeln('住所    =', rec. a);
    writeln('電話    =', rec. tel);
    writeln;
   writeln('新データ入力－>');
    writeln('名前    =');readln(rec. n);
    writeln('住所    =');readln(rec. a);
    writeln('電話    =');readln(rec. tel);
   write('確認 Y/N－>');
   readln(yes);
   if upcase(yes)='Y' then
     begin
      seek(fx, rn-1);
      writeIfx, rec);
     end;
    write('レコード・ナンバー=>');
    readln(rn);
  end;
 close(fx);
end;

procedure apstdfl;
 begin
 end;
procedure rastdfl;
 begin
 end;
procedure rdstdfl;
 begin
 end;

begin
 writeln(レコードの更新……4);
 readln(k);
 case k of
  1: rdstdfl;
  2: rastdfl;
  3: apstdfl;
  4: updstdfl;
 end;
end.
```

(問) プログラム stdfsysl. pas を実行して、住所録ファィルデータを修正しなさい。

3-3. 追加プログラムを手続き化した stdfsys2. pas

住所録に追加するプログラム apdstdfl. pas を手続き化したプログラム stdfsys2. pas についても、前節と全く同じ要領で実行すればよい。

```
program stdfsys2;
 type stdrec=record
            n: string[20];
            a: string[40];
          tal: string[14];
        end;
 var rn: integer;
     fx: file of stdrec;
    rec: stdrec;
    yes: char;
      k: integer;

procedure updstdfl;
 begin
{この部分は同じであるから省略する。}
  end;

procedure apstdfl;
  begin
   assign(fx,'m: stdfile. dat');
   reset(fx);
   seek(fx, filesize(fx));

   write('追加 Y/N －>');readln(yes);
   while(upcase(yes)'='Y') do
  begin
   write('rec. n       =');readln(rec. n);
   write('rec. adress  =');readln(rec. a);
   write('rec. tel     =');readln(rec. tel);
   write(fx, rec);
   write('追加 Y/N－>');readln(yes);
```

```
  end;
 close(fx);
end;

procedure rastdfl;
 begin
 end;
procedure rdstdfl;
 begin
 end;

begin
 writel('新 レコード ヲ ファイル=追加…3');
 writeln('レコード ノ 更新          …4');
 readln(k);
 case k of
  1: rdstdfl;
  2: rastdfl;
  3: apstdfl;
  4: updstdfl;
 end;
end,
```

(問) stdfsys2. pas を実行しなさい。

3-4. ランダム・アクセスを手続き化した stdfsys3. pas

次に、レコード・ナンバーを指定すれば、該当記録を直ちに CRT 画面に書き出すプログラム rasdfl. pas をこのシステムに組み込む場合も、前二者と全く同じようにすればよい。

```
program stdfsys3;
 type stdrec=record
              n: string[20];
              a: string[40];
            tal: string[14];
           end;
 var rn: integer;
     fx: file of stdrec;
     rec: stdrec;
     yes: char;
       k: integer;

procedure updstdfl;
 begin
{この部分は stfsysl. pas とおなじ——省略}
  end;

procedure apstdfl;
 begin
{この部分は stdfsys2. pas と同じ——省略}
  end;

procedure rastdfl;
 begin
  assign(fx,'m: stdfile. dat');
  rese(fx);
  write('レコード・ナンバー=>');
  readln(rn);
  while rn in [1..filesize(fx)] do
   begin
    seek(fx, rn-1);
    read(fx, rec);
    writeln('名前=', rec. n);
    writeln('住所=', rec. a);
    writeln('電話=', rec. tel);
    writeln;
    write('レコード・ナンバー=>');
    readln(rn);
   end;
  close(fx);
end;

procedure rdstdfl;
 begin
 end;

begin
 writeln('Rn 番レコードを CRT に表示…2');
 writeln('新 レコード を ファイルに追加…3');
```

```
 writeln('レコード の 更新                ···4');
 readln(k);
 case k of
  1 : rdstdfl;
  2 : rastdfl;
  3 : apstdfl;
  4 : updstdfl;
 end;
end,
```

(問) stdfsys3. pas を実行して、前節で追加したレコードを表示しなさい。

3-5. プリント出力を手続き化した stdfsys4. pas

この場合は、プログラムの先頭に、uses printer 文を加えるほかは、これまでと同様に処理すればよい。

```
program stdfsys4;
 uses printer;
 type stdrec=record
                n : string[20];
                a : string[40];
              tel : string[14];
             end;
var rn : integer;
    fx : file of stdrec;
   rec : stdrec;
   yes : char;
     k : integer;

procedure updstdfl;
 begin
{stdfsys1. pas と同じ……省略}
  end;

procedure apstdfl;
 begin
 {stdfsys2. pas と同じ……省略}
  end;

procedure apstdfl;
 begin
 {stdfsys3. pas と同じ……省略}
  end;

procedure rastdfl;
 begin
 {stdfsys3. pas と同じ……省略}
  end.

procedure rdstdfl;
 begin
  assign(fx,'m : stdfile. dat');
  reset(fx);
  rn : =1;
  while not eof(fx) do
   begin
    read(fx, rec);
    writeln(lst, 'Rec. No.=', rn);
    writeln(lst, '名前     =', rec. n);
    writeln(lst, '住所     =', rec. a);
    writeln(lst, 'TEL      =', rec. tel);
    writeln(lst);
    rn : =rn+1;
   end;
  close(fx);
 end;
begin
 writeln('全レコード を Prn に出力       ···1');
 writeln('Rn 番レコードを CRT に表示 ···2');
 writeln('新 レコード を ファイルに追加···3');
 writeln('レコード の 更新                ···4');
 readln(k);
 case k of
  1 : rdstdfl;
  2 : rastdfl;
  3 : apstdfl;
  4 : updstdfl;
 end;
end.
```

(問) プログラム stdfsys4. pas の省略部分を完

全にして、stdfsys4. pas を実行しなさい。

3-6. ファイル・オープン手続きの独立と更新手続きの改善

stdfsys1. pas から stdfsys4. pas までのプログラムでは、全ての手続きの最初で

```
assign(fx,'m : stdfile. dat') ;
reset(fx) ;
```

と言う操作を繰り返した。

この住所録ファイルのオープン手続きは共通であるから、

```
procedure fopen ;
 begin
  assign(fx,'m : stdfile. dat') ;
  reset(fx) ;
 end ;
```

として独立させ、各手続きの実行部の最初で

```
fopen ;
```

として、コールすればよいのである。

(問) この方法を採用したプログラム stdfsys5. pas を作成して、実行しなさい。

これまで、レコードを更新するとき、修正する必要の無いものまで全て入力しなければならやかった。これでは不便であるから、変更する必要の無いものは、そのままにし、変更する必要のある部分だけ入力するように修正する必要がある。この点を改良したのが、次に示すプログラムである。

```
write('レコード・ナンバー=>') ; readln(rn) ;
while rn in [1..filesize(fx)] do
 begin
  seek(fx, rn-1) ;
  read(fx, rec) ;
  writeln('Rec. No. =', rn) ;
  writeln('氏名      =', rec. n) ;
  writeln('住所      =', rec. a) ;
  writeln('電話      =', rec. tel) ;
    writeln ;
writeln('データ更新開始') ;
 write(' 氏名確認 Y/N=>') ; readln(yes) ;
 if upcase(yes)<>'Y' then begin
     write('  新氏名     ->') ;
     readln(rec. n) ; end ;
 write('  住所確認 Y/N=>') ;readln(yes) ;
 if upcase(yes)<>'Y' then begin
     write('  新住所     ->') ;
     readln(rec. a) ; end ;
 write(' 電話確認 Y/N=>') ; readln)yes) ;
 if upcase(yes)<>'Y' then begin
     write('  新電話番号->') ;
     readln(rec. tel) ; end ;
 write(' 修正確認 Y/N->') ; readln(yes) ;
 if upcase(yes)'=Y' then
 begin
     seek(fx, rn-1) ;
     write(fx, rec) ;
   end ;
 write('レコード・ナンバー=>') ;
 readln(rn) ;
end ;
```

この方式では、氏名、住所、電話番号のそれぞれについて、確認し、答が" Y" でない時にのみ各項目を入力する。この個別項目のチェックの後、全体について再び確認し、よければファイルに書き戻すのである。

(問) この改良を施したプログラム" stdfsys6. pas" を完成し、実行しなさい。

——以下　次号——